

Bases de Datos Relacionales con MySQL

1 Intro.....	1
2 phpMyAdmin: bienvenida.....	2
3 Crear nuevo usuario.....	3
4 Crear BD.....	4
4.1 Crear la primera tabla: “departamentos”.....	5
4.2 Crear la segunda tabla: “asignaturas”.....	6
4.3 Crear la tercera tabla: “alumnos”.....	7
5 Crear relaciones entre tablas.....	7
6 Introducimos datos en tabla “departamentos”.....	9
7 Introducimos datos en tabla “asignaturas”.....	10
8 Introducimos datos en tabla “alumnos”.....	11
9 Crear una copia de respaldo de tu BDR.....	11
9.1 Recuperar la BDR a partir de una copia de respaldo.....	11
10 Trabajo final BD: “librería”.....	12
10.1 Crear la BD “librería” con MySQL y phpAdmin.....	12
11 OUTRO.....	14

1 Intro

En este tutorial vas a crear **bases de datos relacionales (BDR)** con un servidor MySQL que hay instalado en el Servidor del aula de informática. MySQL es la aplicación que encontrarás normalmente en tu ISP (proveedor de servicios de internet) o en tu proveedor de alojamiento web si tienes algún sitio web. Aunque MySQL tiene su propia interfaz de trabajo, no es muy amigable, por ello preferimos usar una aplicación llamada phpMyAdmin, que nos facilitará enormemente las tareas de crear las bases de datos, sus tablas, insertar, modificar y borrar datos, etc.

En tu PC de casa es muy fácil tener un servidor completo (incluye MySQL, phpMyAdmin, PHP y Apache) tanto para Windows como para Linux, sólo tienes que descargar XAMPP (<http://www.apachefriends.org/es/xampp.html>), e instalarlo, es completamente gratis. En todos nuestros PC's del aula con MAX como S.O. (distro Ubuntu de Linux) ya viene incluidas todas estas aplicaciones típicas de servidores.

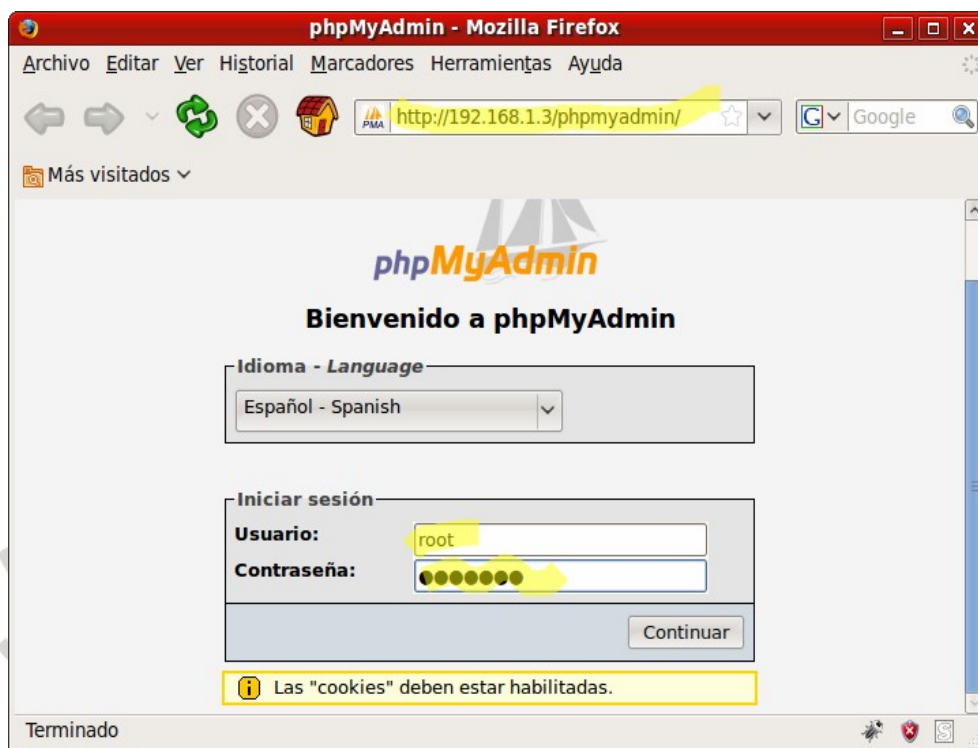
¿Que vamos a aprender nuevo?

1. **Manejo básico de phpMyAdmin**, crearemos BDR, sus tablas, insertaremos nuevos datos, modificaremos otros viejos y borraremos algunos. Crearemos índices para poder establecer relaciones entre tablas y para realizar más eficientemente las consultas y finalmente, crearemos copias de seguridad de nuestra BDR y aprenderemos a restaurarlas.
2. **Comunicación entre PHP y MySQL**. Aprenderemos a crear páginas web con PHP para conectarnos como usuarios autenticados en el servidor MySQL. Luego a seleccionar la BD que queramos y a realizar modificaciones básicas (insertar, modificar, borrar datos...) y sobre todo a realizar CONSULTAS con SQL como aprendimos en el tutorial anterior. Finalmente, aprenderemos a presentar tablas y consultas de nuestra BD en una página Web.

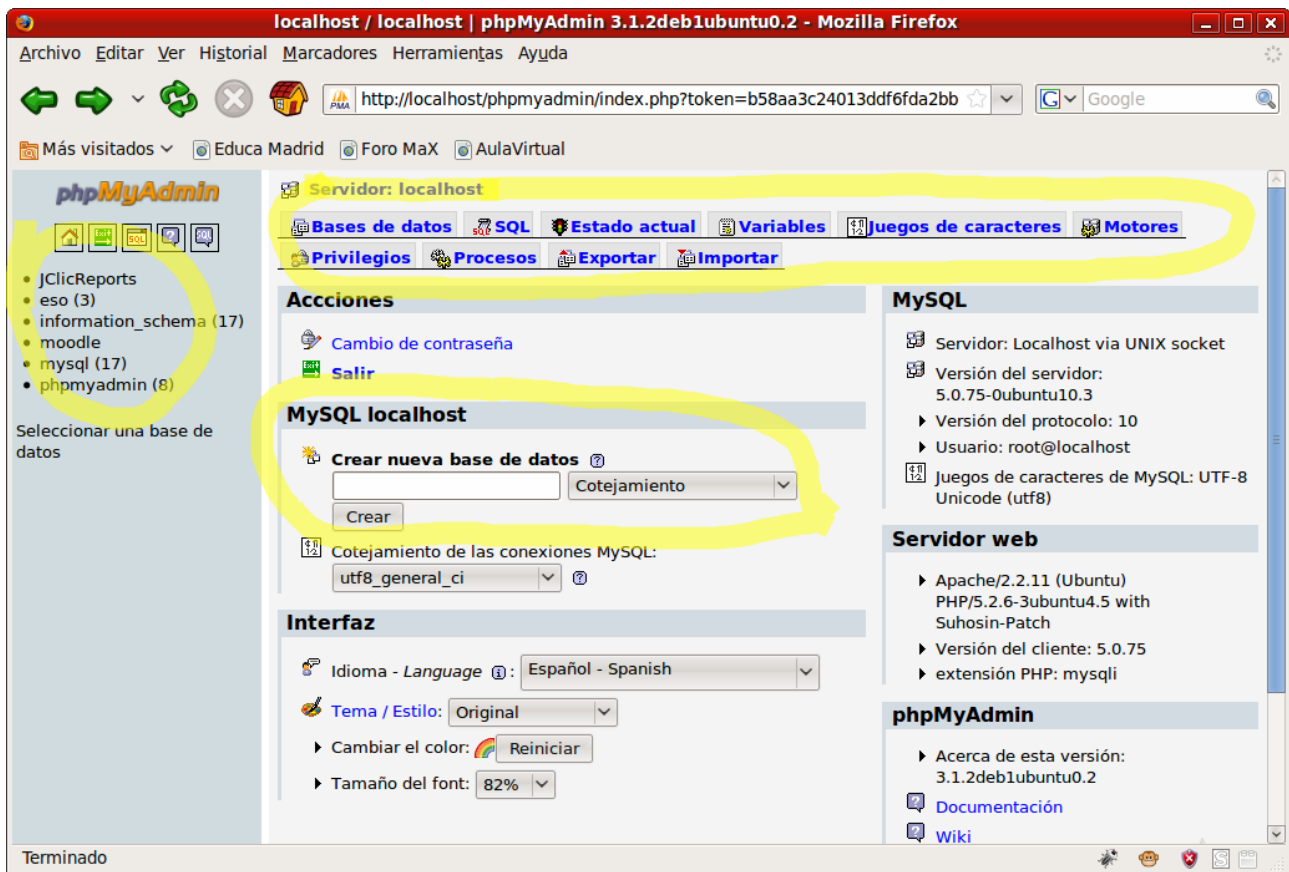
Como ves, vamos a dar un largo paseo...

2 phpMyAdmin: bienvenida

Accedemos a phpMyAdmin desde un navegador web, tecleando en el campo URL, la IP del servidor seguida del nombre de la aplicación, en nuestra aula MAX sería: 192.168.0.116/phpmyadmin. La página principal de phpMyAdmin que aparece es la que muestro abajo. En ella hay que autenticarse (usuario y contraseña) para acceder al gestor de bases de datos MySQL.



El profesor entrará como administrador (root) para crear tu usuario y aparecerá la pantalla de administración de phpMyAdmin que te muestro y explico a continuación:



A destacar marcado en **amarillo** lo siguiente:

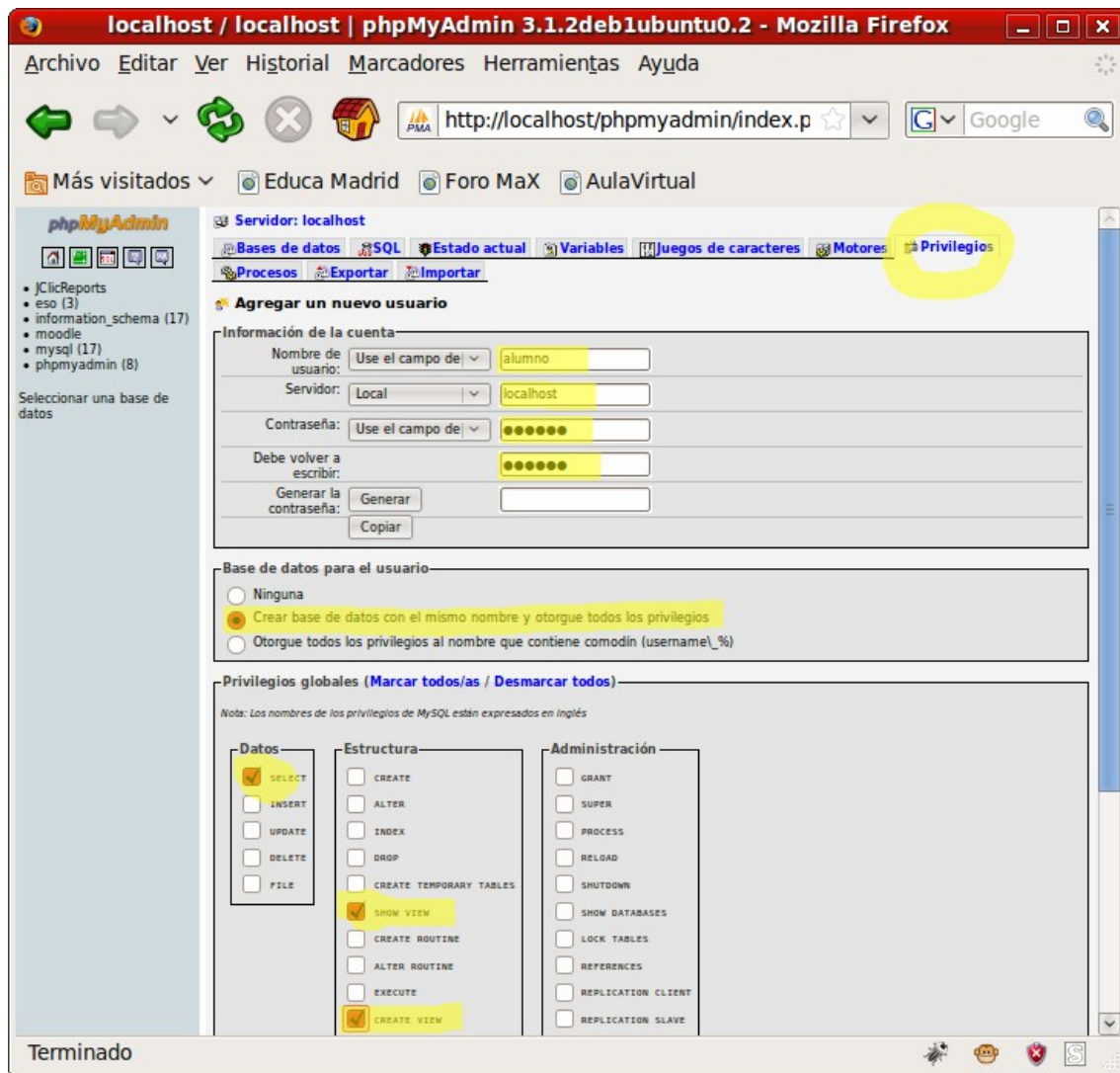
Columna de la Izquierda: bases de datos que hay en el servidor . Algunas han sido creadas por la aplicación (i.e. phpmyadmin), otras (eso) la hemos creado nosotros.

Arriba: Nombre del servidor de BDR, “localhost” y un menú con las herramientas principales. Este menú cambiará según hayamos seleccionado alguna BDR, o alguna tabla.

En Medio: Campo de texto para crear nuestra primera BDR. Además del nombre hay que escoger el juego de caracteres español en la opción cotejamiento.

3 Crear nuevo usuario

1. Pide al profesor que inicie sesión como root (administrador) y en el menú “privilegios” añada un nuevo usuario.
2. Rellena los campos como ves en la siguiente captura, introduce como nombre de usuario “pcx_tunombre_eso”.



3. Le estamos pidiendo que cree una BD nueva con el mismo nombre que el de usuario y que nos otorgue todos los privilegios sobre ella. En cambio, para el resto de BD del servidor sólo nos va a permitir SELECCIONAR Y VISUALIZAR DATOS, pero ninguna modificación es una medida básica de protección...
4. Inicia sesión en phpMyAdmin como el nuevo usuario que has creado y adelante...

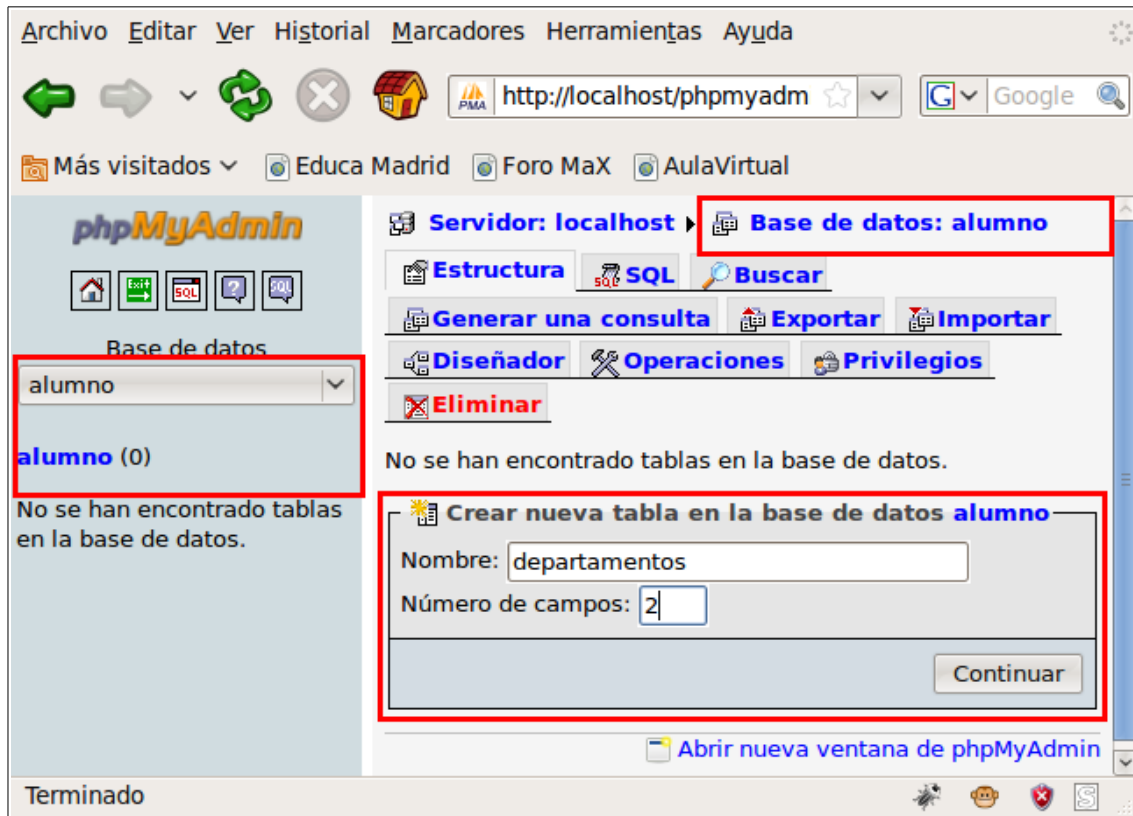
4 Crear BD

Al iniciar sesión como usuario “alumno”, sólo tenemos que seleccionar la BD con el mismo nombre que nuestro usuario, en mi caso “alumno” en la columna de la izquierda.

Nuestra BD tendrá tres tablas, una llamada “departamentos” con sus nombres y los profesores jefes de cada departamento, otra tabla llamada “asignaturas” con las diferentes asignaturas y los departamentos que las imparten. Y la última tabla llamada “alumnos” con los nombres de alumnos y su asignatura preferida.

4.1 Crear la primera tabla: “departamentos”

Empezamos con la tabla “departamentos” que tendrá dos campos. Fíjate en la captura para crear la primera tabla.



1. Ahora debemos definir los dos campos, sus nombres serán “nombre” y “jefe”, que se corresponden con el nombre de cada departamento y con el profesor jefe del departamento, y sus características son:
 - campo “nombre”, tipo de dato es un carácter “varchar” de longitud 150, su juego de caracteres es el español (“latin1_spanish_ci”), no puede ser nulo, y lo más importante, este campo es la clave primaria. Finalmente, su motor de almacenamiento debe ser “InnoDB” que permite definir claves externas (FOREIGN_KEY) en las relaciones entre tablas.
 - campo “jefe”, sus características son iguales al anterior, salvo que NO es la clave primaria.

Si tuviéramos que crear la tabla “departamento” en la base de datos “alumno” y definir sus dos campos mediante lenguaje SQL el código sería este:

```
CREATE TABLE `alumno`.`departamentos` (
  `nombre` VARCHAR( 150 ) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL ,
  `jefe` VARCHAR( 150 ) CHARACTER SET latin1 COLLATE latin1_spanish_ci NOT NULL ,
  PRIMARY KEY ( `nombre` )
) ENGINE = InnoDB;
```

Afortunadamente, phpMyAdmin permite hacer lo mismo de una forma más amigable, sólo tienes que rellenar los datos como se muestra en la captura siguiente.

Servidor: localhost ▶ Base de datos: alumno ▶ Tabla: departamentos

Campo	nombre	jefe
Tipo ?	VARCHAR	VARCHAR
Longitud/Valores* ¹	150	150
Predeterminado ²	Ninguna	Ninguna
Cotejamiento	latin1_spanish_ci	latin1_spanish_ci
Atributos		
Nulo	<input type="checkbox"/>	<input type="checkbox"/>
Índice	PRIMARY	---
A.I.	<input type="checkbox"/>	<input type="checkbox"/>
Comentarios		
MIME-type		
Transformación del navegador		
Opciones de transformación ³		

Comentarios de la tabla:

Motor de almacenamiento: InnoDB

Cotejamiento:

¡No olvides pulsar el botón Grabar situado abajo (no mostrado en la captura), para guardar los nuevos campos!

4.2 Crear la segunda tabla: “asignaturas”

Siguiendo los pasos anteriormente explicados tienes que crear otra tabla:

1. Tabla “asignaturas” con dos campos: nombre de la asignatura y departamento al que pertenece.
2. Campo “nombre”, tipo de dato VARCHAR (150), cotejamiento Latin spanish, no permitir datos nulos, este campo es la CLAVE PRIMARIA y el motor de almacenamiento es InnoDB.
3. Campo “departamento”, con atributos iguales que el campo anterior salvo que NO es clave primaria, sino que es un “Índice” por lo que en el campo Índice=“INDEX”.

La captura siguiente muestra la estructura de la tabla “asignaturas” con sus dos campos definidos y los índices creados...

Servidor: localhost ▶ Base de datos: alumno ▶ Tabla: asignaturas "InnoDB free: 4096 kB"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar

Eliminar

MySQL ha devuelto un valor vacío (i.e., cero columnas). (La consulta tardó 0.0003 seg)

```
SELECT *
FROM "asignaturas"
LIMIT 0 - 38
```

☐ Perfil/Perfilamiento [Editar] [Explicar el SQL] [Crear código PHP] [Actualizar]

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	nombre	varchar(150)	latin1_spanish_ci		No	Ninguna		[iconos]
<input type="checkbox"/>	departamento	varchar(150)	latin1_spanish_ci		No	Ninguna		[iconos]

Marcar todos/as / Desmarcar todos Para los elementos que están marcados: [iconos]

Vista de impresión Vista de relaciones Planteamiento de la estructura de tabla

Añadir 1 campo(s) ● Al final de la tabla ○ Al comienzo de la tabla ○ Después de nombre Continuar

Detalles...

Índices:

Acción	Nombre de la clave	Tipo	Único	Empacado	Campo	Cardinalidad	Cotejamiento	Nulo	Comentario
[iconos]	PRIMARY	BTREE	Sí	No	nombre	0	A		
[iconos]	departamento	BTREE	No	No	departamento	0	A		

Crear un índice en 1 columna(s) Continuar

ATENCIÓN: ¿Para qué vale un **índice**? para encontrar más eficazmente los datos de un campo y para poder establecer relaciones tipo clave externa entre este campo y otros campos indexados de otras tablas. Además, todos los campos definidos como CLAVE PRIMARIA están indexados, por lo que podremos crear relaciones entre ellos sin problemas.

4.3 Crear la tercera tabla: “alumnos”

Siguiendo los mismos pasos, vamos a crear la tabla “alumnos” con sólo dos campos, el nombre del alumno y su asignatura preferida:

Campo “nombre”, tipo de dato VARCHAR (150), cotejamiento Latin_spanish, no permitir datos nulos, este campo es la CLAVE PRIMARIA y el motor de almacenamiento es InnoDB.

Campo “asig_preferida”, con atributos iguales que el campo anterior salvo que NO es clave primaria. También, tenemos que **indexar** este campo, como hicimos con el campo “departamento” de la tabla “asignaturas”.

5 Crear relaciones entre tablas

Establecer relaciones es, sin ninguna duda, una de las herramientas más potentes que tienen las bases de datos, junto con las consultas.

Realmente las relaciones son “restricciones”, que determinan qué datos son válidos en un campo. En general sólo se permitirán insertar en el campo denominado “clave externa” (FOREIGN KEY) aquellos valores que ya existan en otro campo de referencia (REFERENCE). La **tabla “PADRE”** contiene el campo de referencia, mientras que la **tabla “HIJO”** contiene el campo dependiente.

Profundización: un campo **FOREIGN KEY** sólo pueden tener como referencia a campos que sean posean un índice (**PRIMARY KEY, INDEX o UNIQUE**). Por eso en los siguientes ejercicios siempre creamos **INDICES** de los campos que usemos como referencia de otros.

En nuestra BDR vamos a imponer dos restricciones:

1. En la tabla de las “asignaturas”, los nombres de los departamentos deberán existir previamente en la tabla “departamentos”. Es decir, si NO hemos añadido en la tabla “departamentos” el registro “Tecnología”, no podremos en la tabla “asignaturas” asignar la asignatura “TIC” a este departamento.

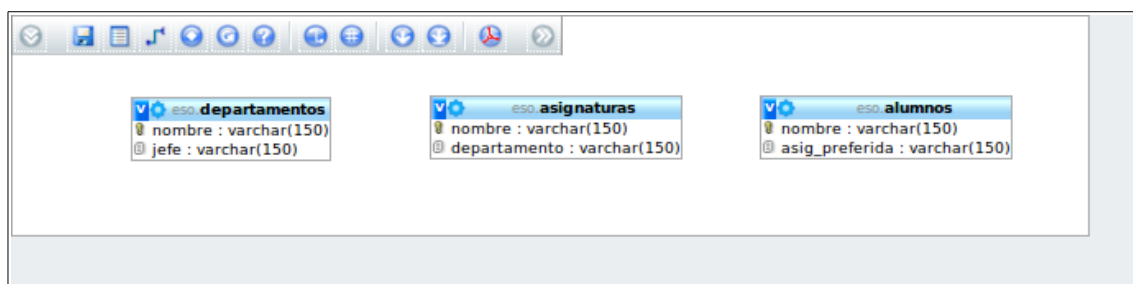
Para crear esta relación con código SQL (pero no lo haremos de esta manera, espera un poco,...):

```
FOREIGN KEY `asignaturas`(`departamento`) REFERENCES `departamentos` (`nombre`);
```

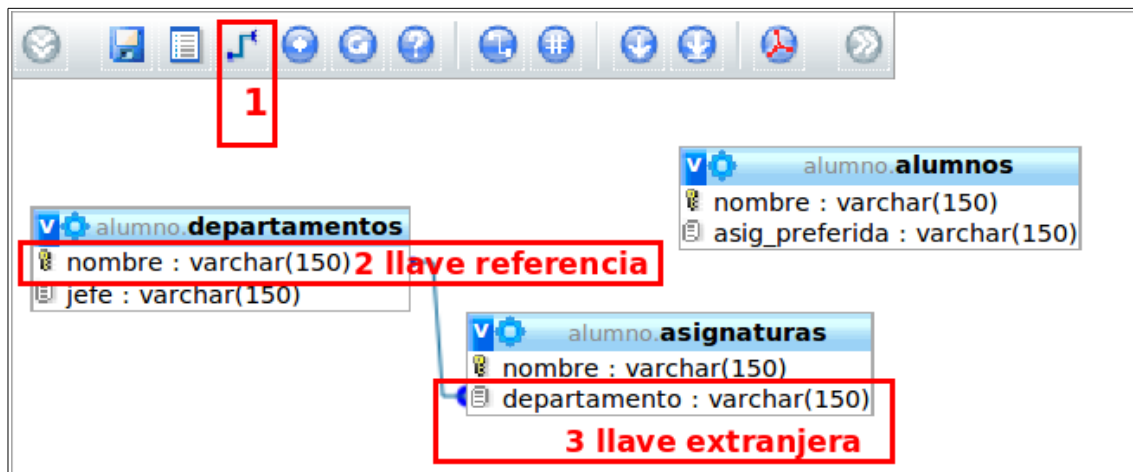
2. En la tabla “alumnos” las asignaturas preferidas de cada alumno deben existir en la tabla “asignaturas”. Es fácil de comprender que si está bien hecha mi BDR ningún alumno pueda elegir una asignatura que no exista en nuestra BDR.

¿Cómo se diseñan las relaciones? Si has comprendido la filosofía de las relaciones, diseñarlas es muy sencillo con phpMyAdmin:

1. Selecciona tu base de datos de la lista que hay en la columna de la izquierda. Luego pulsa en el menú “Relaciones”, verás las tres tablas que has creado como como en la captura:



2. Para crear la relación entre las tablas “departamentos” (será la tabla padre) y “asignaturas” (será la tabla hijo): selecciona una nueva relación (1), luego cuando te pregunte por la llave de referencia (REFERENCE KEY) elige el campo “departamentos.nombre” (2), y como llave extranjera (FOREIGN KEY) el campo “asignaturas.departamento” (3). Cuando te pregunte qué hacer al crear relación FOREIGN_KEY tienes que indicar CASCADE tanto al borrar (ON DELETE) como al actualizar (ON UPDATE). De esta forma al eliminar o alterar algún dato en la tabla padre se modifica igualmente en la tabla hijo.



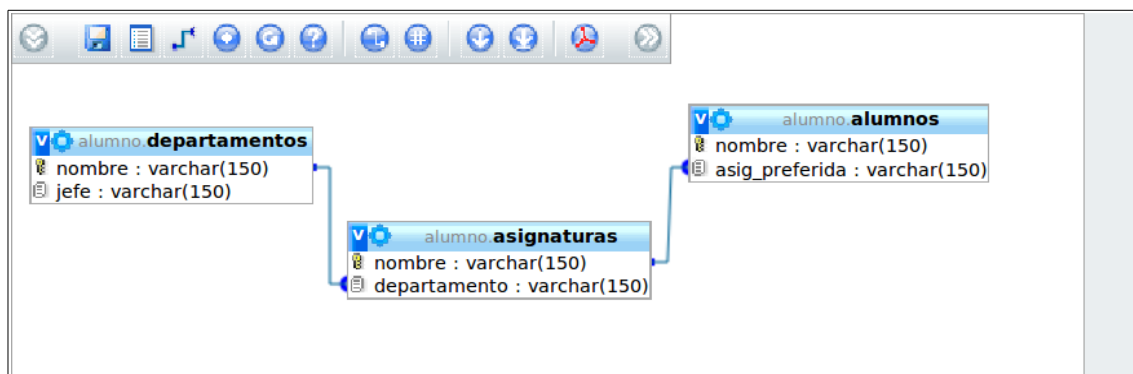
Como resumen final, lo principal de esta relación es:

El campo “asignaturas.departamento” denominado **llave extranjera** sólo podrá tener registros seleccionados de la lista de registros del campo “departamentos.nombre” denominado **clave de referencia**.

Es un poco lioso, pero funciona de maravilla, como veremos después.

Aún nos queda crear la otra relación entre la tabla “asignaturas” (llave de referencia = “asignaturas.nombre”) y la tabla “alumnos” (llave extranjera = “asig_preferida”).

Al final tendremos las dos relaciones definidas tal y como queda en la siguiente captura.



Ahora estamos preparados para introducir los datos en las tres tablas.

6 Introducimos datos en tabla “departamentos”

Esto es realmente sencillo, seleccionamos la primera tabla “departamentos” en la lista de la columna izquierda, luego pulsamos en el menú insertar, rellenamos los dos campos, “nombre” y “jefe” y pulsamos “continuar”...según se muestra en la captura.

Servidor: localhost Base de datos: alumno Tabla: departamentos "InnoDB free: 4096 kB"

Examinar Estructura SQL Buscar Insertar Exportar Importar Operaciones Vaciar

Eliminar

Campo	Tipo	Función	Nulo	Valor
nombre	varchar(150)			Ciencias Naturales
jefe	varchar(150)			Alejandro

Continuar

☒ Ignorar

Campo	Tipo	Función	Nulo	Valor
nombre	varchar(150)			

Al final debemos rellenar los siguientes datos en esta tabla "departamentos"

nombre	jefe
Ciencias Naturales	Alejandro
Ciencias Sociales	Mauricio
Economía	Ángeles
Física y Química	Ricardo
Francés	Sara
Girego	Rufi
Inglés	Inmaculada
Lengua y Literatura Castellana	Cuca
Matemáticas	Sagrario
Religión	Paulo
Tecnología	Javier

Tabla "departamentos"

7 Introducimos datos en tabla "asignaturas"

Los datos son los siguientes:

nombre	departamento
inglés	Inglés
teatro	Lengua y Literatura Castellana
refuerzo de matemáticas	Matemáticas
IVL	Tecnología
tecnología	Tecnología
TIC	Tecnología

Tabla "asignaturas"

¡¡¡Atento!!! Si las relaciones entre las tablas están bien hechas MySQL sólo te permitirá seleccionar los nombres de los departamentos de entre una lista propuesta...si no es así, ¡ llama a tu profesor porque has definido mal las relaciones !

8 Introducimos datos en tabla “alumnos”

Los datos son los siguientes:

nombre ▲	asig_preferida
Alumno 1	IVL
Alumno 2	tecnología
Alumno 3	TIC
Alumno 4	refuerzo de matemáticas
Alumno 5	teatro

tabla "alumnos"

Vuelve a probar la integridad referencial de tus datos...

Finalmente, si las tres tablas de tu BDR están bien hechas su estructura debe coincidir con la mostrada en esta captura.

Servidor: localhost ▶

Base de datos: eso

Estructura

SQL

Buscar

Generar una consulta

Exportar

Importar

Diseñador

Operaciones

Privilegios

Eliminar

	Tabla	Acción	Registros ¹	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/>	alumnos	<div><div></div><div></div><div></div><div></div><div></div><div></div></div>	5	InnoDB	latin1_spanish_ci	32.0 KB	-
<input type="checkbox"/>	asignaturas	<div><div></div><div></div><div></div><div></div><div></div><div></div></div>	6	InnoDB	latin1_spanish_ci	32.0 KB	-
<input type="checkbox"/>	departamentos	<div><div></div><div></div><div></div><div></div><div></div><div></div></div>	11	InnoDB	latin1_spanish_ci	16.0 KB	-
	3 tabla(s)	Número de filas	22	MyISAM	latin1_swedish_ci	80.0 KB	0 Bytes

10 Trabajo final BD: “librería”

10.1 Crear la BD “librería” con MySQL y phpAdmin

- Usando el mismo usuario y la misma BD que ya tenías.
- Vas a diseñar una base de datos para controlar una “librería”. Tendrás que añadir varias tablas con los siguientes campos y atributos.
 - Tabla “autor”, en la tabla te muestro los atributos de los campos:

Tabla “autor”, motor de almacenamiento InnoDB	
Campo 1	`autor_nombre` varchar(150) collate latin1_spanish_ci NOT NULL, PRIMARY KEY
Campo 2	`autor_nacionalidad` varchar(50) collate latin1_spanish_ci NOT NULL

Quedaría como se ve a continuación en phpmyadmin:

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/>	autor_nombre	varchar(150)	latin1_spanish_ci		No	None		[Icons]
<input type="checkbox"/>	autor_nacionalidad	varchar(50)	latin1_spanish_ci		No	None		[Icons]

- Tabla “libro”, en la tabla te muestro los atributos de los campos:

Tabla “libros”, motor de almacenamiento=InnoDB	
Campo 1	`libro_id` int(11) NOT NULL auto_increment, PRIMARY KEY
Campo 2	`libro_autor` varchar(150) collate latin1_spanish_ci NULL, INDEX.
Campo 3	`libro_isbn` varchar(125) collate latin1_spanish_ci NOT NULL, COMMENT 'código internacional del libro', INDEX.
Campo 4	`libro_titulo` varchar(150) collate latin1_spanish_ci default NULL, INDEX.
Campo 5	`libro_editorial` varchar(125) collate latin1_spanish_ci default NULL, INDEX
Campo 6	`libro_paginas` int(20) default NULL, COMMENT 'numero de páginas

	del libro',
Campo 7	`libro_encuadernacion` enum('Rústica','Pasta dura','Bolsillo') collate latin1_spanish_ci default NULL
Campo 8	`libro_año` int(20) NOT NULL COMMENT 'Año de Publicación del Libro'

Quedaría como se ve a continuación en phpmyadmin:

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
<input type="checkbox"/>	libro_id	int(11)			No	None	auto_increment
<input type="checkbox"/>	libro_autor	varchar(150)	latin1_spanish_ci		Sí	NULL	
<input type="checkbox"/>	libro_isbn	varchar(125)	latin1_spanish_ci		No	None	
<input type="checkbox"/>	libro_titulo	varchar(150)	latin1_spanish_ci		Sí	NULL	
<input type="checkbox"/>	libro_editorial	varchar(125)	latin1_spanish_ci		Sí	NULL	
<input type="checkbox"/>	libro_paginas	int(20)			Sí	NULL	
<input type="checkbox"/>	libro_encuadernacion	enum('Rústica','Pasta dura','Bolsillo')	latin1_spanish_ci		Sí	NULL	
<input type="checkbox"/>	libro_año	int(20)			No	None	

- 2.3. Tabla “cliente”. Tendrá tres campos, el nombre completo del cliente, su dirección, y su edad. Te dejo que pienses en los atributos y los definas con sentido común.
- 2.4. Tabla ‘carrito’. Tendrá un campo que identifique cada compra “carrito_id”, que será la llave primaria. Otro campo mostrará el título del libro vendido, “carrito_libro” y un tercer campo el nombre del usuario comprador, “carrito_cliente”.
- 2.5. Establece las relaciones entre las campos de las tablas que creas precisas.

Pista: un ejemplo de relación sería que en la tabla “carrito” no se pueda vender un libro que no tengamos en la librería, es deci, en la tabla “libros”.

- 2.6. Introduce los datos de la tabla “autor” que ves a continuación, o mejor, cópialos desde el archivo **db_libreria_2.ods** que encontrarás junto a este archivo pdf que estás leyendo.

autor_nombre	autor_nacionalidad
Muriel Barbery	Marruecos
Paolo Giordano	Italia
Agota Kristof	Hungría
José Saramago	Portugal
Germán Castro Caycedo	Colombia
Mark Haddon	Reino Unido
Terenci Moix	España
Gabriel García Márquez	Colombia
Stieg Larsson	Suecia
Josh Bazell	USA

- 2.7. Introduce los datos de la tabla “libro” que ves a continuación, estos datos están tb. en el archivo **db_libreria_2.ods**:

libro_id	libro_autor	libro_isbn	libro_título	libro_editorial	libro_paginas	libro_encuadernacion	libro_año
1	Muriel Barbery	978-84-322-2811-6	La elegancia del erizo	Seix Barral	364	Rústica	2009
2	Paolo Giordano	978-84-9838-205-1	La soledad de los números primos	Salamandra	281	Rústica	2009
3	Agota Kristof	978-84-9711-084-6	Claus y Lucas	El Aleph	444	Bolsillo	2009
5	José Saramago	204-7463-2	El viaje del elefante	Alfaguara	270	Pasta dura	2009
6	Germán Castro Caycedo	958-614-650-0	Perdido en el Amazonas	Planeta	281	Rústica	2001
7	Mark Haddon	978-84-7888-910-5	El curioso incidente del perro a medianoche	Salamandra	268	Rústica	2008
8	Terenci Moix	84-473-0126-5	No digas que fué un sueño	RBA	378	Pasta dura	1993
9	Gabriel García Márquez	84-473-0121-4	El general en su laberinto	RBA	285	Pasta dura	1993
10	Gabriel García Márquez	84-473-1514-2	Noticia de un secuestro	RBA	254	Pasta dura	1999
11	Stieg Larsson	978-84-233-4044-6	Los hombres que no amaban a las mujeres	Destino	665	Rústica	2009
12	Josh Bazell	978-84-339-7512-6	Burlando a la parca	Anagrama	316	Rústica	2009

- 2.8. En la tabla “cliente”, introduce 10 registros cualesquiera.
- 2.9. En la tabla “carrito”, introduce 15 compras. Intenta que algún libro se venda varias veces, y que algún cliente realice varias compras.
- 2.10. Comprueba que funcionan las relaciones establecidas entre las tablas.
- 2.11. Haz una copia de respaldo de la “db_librería” y envíasela a tu profesor.

11 OUTRO

Bien por fin has acabado... en este tutorial has aprendido a trabajar con el gestor de bases de datos MySQL. En la siguiente entrega abordaremos cómo realizar consultas SQL, dentro de phpMyAdmin o desde una página web externa con PHP.